Appendix E

WSBen Manual

E.1 Motivation

The database and AI community increasingly devotes attention to discovery and composition problem of large volumes of web services. As many prototypes of dedicated web services discovery and composition tools are available, there is a strong need for a framework to analyze the capabilities and performance of such tools as early as possible. Thus, WSBen is designed and developed with aiming at providing flexible benchmarks that helps users and developers to gain insights into the characteristics of their algorithms and products.

E.2 Purpose of WSBen

WSBen is a web services generator which produces scaled WSDL documents with user-defined characteristics and sample queries to run a user’s proposal. Both documents of WSDL and sample queries are according to the DTD specified in the web service challenges (EEE05 and ICEBE05). In addition, WSBen allows users to translate WSDL documents into a PDDL or Strips file so that they can compare their tools with existing AI planners concurrently.
E.3 How to use WSBen

E.3.1 Installation

- WSBen requires that Python version 2.3 or greater be installed.
- WSBen requires NetworkX\(^1\) that is a Python package for the creation, manipulation, as well as the study of the structure, dynamics, and functions of complex networks.
- WSBen is downloadable from the official WSBen homepage [111].

E.3.2 Options

WSBen provides two input parameter frameworks: \(xTS\) and \(yTS\). They are different from each other in terms of their approaches to specify the parameter cluster network. In this manual, we assume that users choose to use \(xTS\) because of its easiness to use. WSBen comes with a number of options to influence the output behavior. Users are first recommended to understand our framework to generate web services described in the full version of technical reports which are available in our official WSBen homepage.

- \(-j\, ⟨factor⟩\): The number of clusters (jars).
- \(-t\, ⟨factor⟩\): The (total) number of web services to produce.
- \(-r\, ⟨factor⟩\): The parameter condense rate. This value is used to specify the size of cluster.
- \(-m\, ⟨factor1, factor2, factor3, factor4⟩\): The graph model of the parameter cluster network. \(factor1\) must be one among “ba”, “nws”, and “er”. If \(Factor1\) is set to “ba”, then \(Factor2\) and \(Factor3\) are assigned for \(N\) and \(m_0\) of the Barabasi-Albert scale-free network model, respectively; \(Factor4\) is not required. If \(Factor1\) is set to “nws”, then \(Factor2\), \(Factor3\), and \(Factor4\) are assigned for \(N\), \(k\), and \(p\) of the Newman-Watts-Strogatz small-world network.

\(^1\)https://networkx.lanl.gov/
model, respectively. If Factor1="er", then Factor2 and Factor 3 are assigned for N and p of the Erdos-Renyi random graph model, respectively.

- -o \{out_name\}: It is used to specify the name of the output directory in which the generated WSDL files are located.

- -q: It generates a query file. Default name is "\{out_name\}.query".

- -a: It translates WSDL files into a PDDL and a Strips file. Default names are "\{out_name\}.pddl" and "\{out_name\}.strips".

- -p: It generates a report where a parameter and its usage frequency are described comma-separately. Default name is "\{out_name\}_parDegree.txt". This file can be read by MS-Excel as an csv file format.

### E.3.3 Basic usage (examples)

In the following examples, WSBen will generate three test sets based on different parameter cluster networks as follows:

- Scale-free network:
  
  >python WSBen.py -j 100 -t 1000 -r 1 -m ba,100,6 -o ./ba/ -q -a -p

- Small-world network:
  
  >python WSBen.py -j 100 -t 1000 -r 1 -m nws,100,6,0.1 -o ./nws/ -q -a -p

- Random graph:
  
  >python WSBen.py -j 100 -t 1000 -r 1 -m er,100,0.06 -o ./er/ -q -a -p

Each test set illustrated above can be downloaded from the WSBen official homepage. Further documentations, such as the full version of technical reports and the presentation file of WSBen system, are downloadable from the WSben official homepage too.

### E.3.4 GUI version

WSBen is also provided in a GUI-version, as shown in Figure E.1. The widgets in the GUI, such as buttons and check buttons play the same roles as the command
Figure E.1: GUI version of WSBen

options in the previous console version. One added function to the GUI version is that users can see the parameter cluster network formed by setting $xTS$ by values. In Figure E.1, the three networks below the GUI are the sample parameter cluster networks, where each circular node represent a cluster and edges with heads denote the web service template, from which web services are instanced. The size of node is proportional to the number of parameters in the node, while the transparency level of a node’s color is inversely proportional to the degree of the node. For example, the hub cluster in a parameter cluster network that is characterized by the high degree and small number of parameters is presented by a small circle with less transparent color in the graph.

E.4 Trouble Shooting

If WSBen does not show any message or if it returns errors, make sure Python was installed correctly and the executable was added to your Path system environment variable. If everything looks alright with your Path, make sure you are in the
directory of the WSBen.py file. Any other problem please contact us through the WSBen official homepage.